

## Use s3cmd with rados gateway.

For use s3cmd with our cluster, please refer to attached s3 config file.

1. In this configuration, we assign the gw ip in configuration file so need not to setup or reference to dns server
2. For ex. If 192.168.1.161 is mars200 radosgw ip

Following setting in configuration should be change according to your keys and rados gateway IP:

```
access_key = YOUR_ACCESS_KEY
secret_key = YOUR_SECRET_KEY
host_base = 192.168.1.161:7480
host_bucket = 192.168.1.161:7480/%\(bucket\)
use_https = False
```

Using this configuration file you can use

"s3cmd mb [s3://YOUR\\_BUCKET\\_NAME](#)" to create bucket

And for ExpanDrive,  
Remember to put "YOUR\_BUCKET\_NAME" you created in "remote path" entry.

[default]

```
access_key = YOUR\_ACCESS\_KEY
access_token =
add_encoding_exts =
add_headers =
bucket_location = US
ca_certs_file =
cache_file =
check_ssl_certificate = True
check_ssl_hostname = True
cloudfront_host = cloudfront.amazonaws.com
default_mime_type = binary/octet-stream
delay_updates = False
delete_after = False
delete_after_fetch = False
delete_removed = False
dry_run = False
```

enable\_multipart = True  
encoding = UTF-8  
encrypt = False  
expiry\_date =  
expiry\_days =  
expiry\_prefix =  
follow\_symlinks = False  
force = False  
get\_continue = False  
gpg\_command = None  
gpg\_decrypt = %(gpg\_command)s -d --verbose --no-use-agent --batch --yes --passphrase-fd %(passphrase\_fd)s -o %(output\_file)s %(input\_file)s  
gpg\_encrypt = %(gpg\_command)s -c --verbose --no-use-agent --batch --yes --passphrase-fd %(passphrase\_fd)s -o %(output\_file)s %(input\_file)s  
gpg\_passphrase =  
guess\_mime\_type = True  
host\_base = 192.168.1.161:7480  
host\_bucket = 192.168.1.161:7480/%(bucket)  
human\_readable\_sizes = False  
invalidate\_default\_index\_on\_cf = False  
invalidate\_default\_index\_root\_on\_cf = True  
invalidate\_on\_cf = False  
kms\_key =  
limit = -1  
limitrate = 0  
list\_md5 = False  
log\_target\_prefix =  
long\_listing = False  
max\_delete = -1  
mime\_type =  
multipart\_chunk\_size\_mb = 15  
multipart\_max\_chunks = 10000  
preserve\_attrs = True

progress\_meter = True  
proxy\_host =  
proxy\_port = 0  
put\_continue = False  
recursive = False  
recv\_chunk = 65536  
reduced\_redundancy = False  
requester\_pays = False  
restore\_days = 1  
restore\_priority = Standard  
secret\_key = YOUR\_SECRET\_KEY  
send\_chunk = 65536  
server\_side\_encryption = False  
signature\_v2 = False  
simplifiedb\_host = sdb.amazonaws.com  
skip\_existing = False  
socket\_timeout = 300  
stats = False  
stop\_on\_error = False  
storage\_class =  
urlencoding\_mode = normal  
use\_http\_expect = False  
use\_https = False  
use\_mime\_magic = True  
verbosity = WARNING  
website\_endpoint = http://%(bucket)s.s3-website-%(location)s.amazonaws.com/  
website\_error =  
website\_index = index.html

